Using DBMS_ADVANCED_REWRITE When Binds Are Present (Avoiding ORA-30353) (Doc ID 392214.1)

In this Document

APPLIES TO:

Oracle Server - Enterprise Edition - Version 10.1.0.2 to 11.2.0.2 [Release 10.1 to 11.2]
Information in this document applies to any platform.

GOAL

When trying to use DBMS_ADVANCED_REWRITE to rewrite a query with bind variables in the where clause, the error ORA-30353 is signalled.
This is a known and intended limitation in the functionality.

A rewrite of the query in spite of this limitation can still be accomplished, as shown in this article.

Here is a small testcase to demonstrate the above. It is assumed that the prerequisites for rewrite equivalence are satisfied in the current session i.e. user has execute privilege on DBMS_ADVANCED_REWRITE, query_rewrite_enabled=true and query_rewrite_integrity=trusted (or stale_tolerated:)

```
SQL> CREATE TABLE tableA (c1 VARCHAR2(10));

Table created.

SQL> INSERT INTO tableA VALUES ('A');

1 row created.

SQL> INSERT INTO tableA VALUES ('C');

1 row created.

SQL> CREATE TABLE tableB (c1 VARCHAR2(10));

Table created.

SQL> INSERT INTO tableB VALUES ('B');

1 row created.

SQL> INSERT INTO tableB VALUES ('C');

1 row created.

SQL> COMMIT;

Commit complete.
```

Consider the following query and its result without rewrite equivalence:

```
SQL> variable B1 varchar2(1)
SQL> exec :B1 := 'C';

PL/SQL procedure successfully completed.

SQL> SELECT c1 FROM tableA WHERE c1 != :B1;

C1
----------
```

```
A
```

We wish to transform it so that it selects from tableB instead:

```
SQL> exec sys.dbms_advanced_rewrite.declare_rewrite_equivalence ( -
> name => 'DUMMY', -
> source_stmt => 'SELECT c1 FROM tableA where c1 != :B1', -
> destination_stmt => 'SELECT c1 FROM tableB', -
> validate => FALSE, -
> rewrite_mode => 'GENERAL' -
> );

BEGIN sys.dbms_advanced_rewrite.declare_rewrite_equivalence ( name => 'DUMMY',
source_stmt => 'SELECT c1 FROM tableA where c1 != :B1', destination_stmt =>
'SELECT c1 FROM tableB', validate => FALSE, rewrite_mode => 'GENERAL' ); END;
*
ERROR at line 1:
ORA-30353: expression not supported for query rewrite
ORA-06512: at "SYS.DBMS_ADVANCED_REWRITE", line 29
ORA-06512: at "SYS.DBMS_ADVANCED_REWRITE", line 185
ORA-06512: at line 1
```

The message for ORA-30353 is:

```
30353, 00000, "expression not supported for query rewrite"
// *Cause: The select clause referenced UID, USER, ROWNUM, SYSDATE,
//         CURRENT_TIMESTAMP, MAXVALUE, a sequence number, a bind variable,
//         correlation variable, a set result,a  trigger return variable, a
//         parallel table queue column, collection iterator, etc.
//
// *Action: Remove the offending expression or disable the REWRITE option on
//          the materialized view.
```

FIX

To accomplish this, use DBMS_ADVANCED_REWRITE to rewrite the sql statement without specifying the where clause and use REWRITE_MODE=>'GENERAL'. Use caution since this more general approach may cause other queries to be rewritten as well.

```
SQL> exec sys.dbms_advanced_rewrite.declare_rewrite_equivalence ( -
> name => 'DUMMY', -
> source_stmt => 'SELECT c1 FROM tableA', -
> destination_stmt => 'SELECT c1 FROM tableB', -
> validate => FALSE, -
> rewrite_mode => 'GENERAL' -
> );

PL/SQL procedure successfully completed.
```

Having done the above, the original query using the Bind Variable is transformed to work against tableB instead of tableA:

```
SQL> variable B1 varchar2(1)
SQL> exec :B1 := 'C';

PL/SQL procedure successfully completed.

SQL> SELECT c1 FROM tableA WHERE c1 != :B1;

C1
----------
B
```

REFERENCES

BUG:4330896 - CAN'T USE DBMS_ADVANCED_REWRITE WHEN SQL TEXT HAS BINDS ORA-30389

Didn't find what you are looking for?